2152

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#3/an
10-10-01

In re U.S. Patent Application of

MACIEL

Application Number: To Be Assigned 09918639

Filed: Concurrently Herewith

For: METHOD OF TRANSFERRING DATA BETWEEN MEMORIES OF COMPUTERS

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

NOTICE OF PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of January 12, 2001, the filing date of the corresponding Japanese patent application 2001-004399.

The certified copy of corresponding Japanese patent application 2001-004399 is being submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,

Stanley P. Fisher
Registration Number 24,344

JUAN CARLOS A. MARQUEZ
Registration No. 34,072

REED SMITH HAZEL & THOMAS LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
September 11, 2001

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　１月１２日

出願番号
Application Number: 特願２００１－００４３９９

出願人
Applicant(s): 株式会社日立製作所

２００１年　７月２７日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

出証番号　出証特２００１－３０６５３６３

【書類名】　特許願

【整理番号】　H00022231A

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 13/00

【発明者】

【住所又は居所】　東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社日立製作所中央研究所内

【氏名】　マシエル　フレデリコ

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社　日立製作所

【代理人】

【識別番号】　100075096

【弁理士】

【氏名又は名称】　作田　康夫

【電話番号】　03-3212-1111

【手数料の表示】

【予納台帳番号】　013088

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　通信方法

【特許請求の範囲】

【請求項１】

通信手段を介して情報処理装置間でデータを転送する通信方法であり、

受信側となるべき第一の情報処理装置は送信側となるべき第二の情報処理装置に対し、前記データの受信対象のメモリ領域を連絡するようにされた通信方法において、

前記第一の情報処理装置は前記第二の情報処理装置に対し、前記受信対象のメモリ領域を第二の情報処理装置から指示して該受信対象のメモリ領域に前記データを転送する第一の転送動作と、前記第一の情報処理装置に予め割り振ったバッファ領域を介して該データを転送する第二の転送動作との何れを選択すべきかを判定するための転送データ長に関する閾値を通知することを特徴とする通信方法。

【請求項２】

前記閾値は転送のスループットを向上するために定められることを特徴とする請求項１の通信方法。

【請求項３】

前記閾値は転送のレイテンシーを削減するために定められることを特徴とする請求項１の通信方法。

【請求項４】

前記閾値は転送の処理量を削減するために定められることを特徴とする請求項１の通信方法。

【請求項５】

通信手段を介して情報処理装置間でデータを転送する通信方法であり、

受信側となるべき第一の情報処理装置は送信側となるべき第二の情報処理装置に対し、前記データの受信対象のメモリ領域を連絡するようにされた通信方法において、

前記第一の情報処理装置は前記第二の情報処理装置に対し、前記受信対象のメモリ領域を第二の情報処理装置から指示して該受信対象のメモリ領域に前記データ

を転送する第一の転送動作と、前記第一の情報処理装置に予め割り振ったバッファ領域を介して該データを転送する第二の転送動作との何れを選択すべきかを判定するための転送データ長に関する閾値を通知し、
前記第二の情報処理装置は転送すべきデータ長が前記閾値を越えるか否かにより前記第一の転送動作か、第二の転送動作かを決定して前記データを転送することを特徴とする通信方法。

【請求項６】

通信手段に接続し、上記の通信手段を介して第二情報処理装置からデータを受信し、上記の通信手段でデータを受信する前に対象のメモリ領域を受信可能な領域として指示する第一の情報処理装置において、
上記受信可能な領域として指示する動作の処理時間により、あらかじめ割り振って指示したメモリ領域の大きさを決定し、上記のメモリ領域の大きさを上記第二情報処理装置に知らせ、
第二情報処理装置に上記メモリ領域の大きさを超えないデータ長の送信を上記あらかじめ割り振って指示したメモリ領域に送信してもらい、超えるデータ長の送信に対象のメモリ領域を指示して上記対象のメモリ領域に送信してもらうことにより、最速の通信方法を使用することを特徴とする通信方法。

【請求項７】

通信手段を介して情報処理装置間でデータを転送する通信方法であり、
受信側となるべき第一の情報処理装置は前記データの受信対象のメモリ領域を登録し、
前記受信対象のメモリ領域のアドレスを送信側となるべき第二の情報処理装置に対して通知することを特徴とする通信方法。

【請求項８】

前記第一の情報処理装置は、前記受信対象のメモリ領域の登録が必要か否かを判定し、必要があった時にのみ前記メモリ領域の登録と前記アドレスの第二の情報処理装置に対する通知とを実行することを特徴とする請求項７の通信方法。

【請求項９】

前記判定は前記アドレスの通知の効率を測定することにより実行することを特徴

とする請求項８の通信方法。

【請求項１０】

通信手段に接続し、上記の通信手段を介して第二情報処理装置にデータを送信し、上記の通信手段でデータを送信する前に対象のメモリ領域を送信可能な領域として指示する第一の情報処理装置において、

あらかじめ割り振って指示したメモリ領域に送信データをコピーし、上記コピーしたデータのアドレスとデータ量を上記第二情報処理装置に知らせ、上記第二情報処理装置にデータを読み出すことを特徴とする通信方法。

【請求項１１】

通信手段に接続し、上記の通信手段を介して第二情報処理装置にデータを送信し、上記の通信手段でデータを送信する前に対象のメモリ領域を送信可能な領域として指示する第一の情報処理装置において、

あらかじめ割り振って指示したメモリ領域に送信データをコピーし、上記コピーしたデータを、上記第二情報処理装置がこの通信に指示したメモリ領域に送信することを特徴とする通信方法。

【請求項１２】

通信相手のメモリアドレスを指定しデータを送信できる通信手段に接続し、上記の通信手段を介して第二情報処理装置からデータを受信する第一の情報処理装置において、

第二情報処理装置がこのデータ転送に指示しアドレスとデータ量を知らせたメモリ領域から、第一情報処理装置があらかじめ割り振って指示したメモリ領域に読み出すことを特徴とする通信方法。

【請求項１３】

通信手段に接続し、上記の通信手段を介して複数のデータ転送方法を持つ通信プロトコルで送受信する第一と第二の情報処理装において、

送受信開始時、第一およびまたは第二の情報処理装が通信相手に平均転送データ長を知らせ、

上記平均転送データ長により転送方法を選択することを特徴とする通信方法。

【請求項１４】

上記転送方法の選択は、対象のメモリ領域を指示して送受信するか否か、およびまたはあらかじめ割り振って指示したメモリ領域を介してデータを送受信するか否かを特徴とする請求項１３の通信方法。

【請求項１５】

通信相手のメモリアドレスを指定しデータを送受信できる通信相手のメモリアドレスを指定しデータを送信できる通信手段に接続し、上記の通信手段を介して第二情報処理装置とデータを送受信し、上記の通信手段でデータを受信する前に対象のメモリ領域を受信可能な領域として指示し、あらかじめ割り振って指示したメモリ領域を介してデータを送受信する第一の情報処理装置において、

上記あらかじめ割り振って指示したメモリ領域を変更することを特徴とする通信方法。

【請求項１６】

上記変更が上記メモリ領域の拡大およびまたは縮小であることを特徴とする請求項１５の通信方法。

【請求項１７】

上記あらかじめ割り振って指示したメモリ領域は受信用途と受信用途に分かれおり、上記変更が受信用途のメモリ領域を送信用途にすること、およびまたは送信用途のメモリ領域を受信用途にすることを特徴とする請求項１５の通信方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、複数の種類の通信網により接続された複数の計算機を有する計算機システムにおける、計算機間のデータ送受信方法に係り、特に計算機間メモリ間データ転送の機能を持つネットワークとハードウェアの上での計算機間データ送受信方法に関する。

【０００２】

【従来の技術】

計算機間通信、特にインターネットやイントラネットでの通信には、ＴＣＰ／ＩＰプロトコルが極めて一般的に使用されている。ＴＣＰ／ＩＰ処理をアプリケ

ーションでなくオペレーティングシステムが行うため、アプリケーションがＴＣＰ／ＩＰで通信するために「ソケット」と呼ばれるＡＰＩ（Application Programming Interface、アプリケーションがコンピュータやオペレーティングシステムのある機能を用いるために呼び出す関数の集合）を用いる（W. Richard Stevens, "UNIX Network Programming," Prentice Hall, U.S.A., 1990, ISBN 0-13-949876-1参照）。

【０００３】

図１にＴＣＰ／ＩＰプロトコルを使用し通信するホストのソフトウェア構成例を示す。ホスト１０はネットワーク１８を使用して通信する。ホスト１０のオペレーティングシステムのカーネル１２０がＴＣＰ／ＩＰのプロトコル処理１２１をし、通信ハードウェア１１を制御して通信する。アプリケーション１００のプログラム１０１がソケットＡＰＩ９０を用い、ライブラリ１１０を呼び出す。ライブラリがシステムコール１１１を実行してカーネル１２０を呼び出す。カーネル１２０がソケット用バッファ１２２を介して、アプリケーション１００のデータ１０２を送受信する。

【０００４】

ＴＣＰ／ＩＰ通信はプロトコル処理１２１の処理量が多く、そしてシステムコール１１１と、データ１０２とソケットバッファ１２２の間のコピーはオーバーヘッドとなるため、これらの処理は通信性能を制限することがある。このため、スーパーコンピュータやワークステーションクラスタのような、高速な通信を必要とする計算機システムでは、プロトコル処理、システムコールとデータコピーをせず、カーネルを介さずにアプリケーション間データ転送ができるネットワークが用いられる。本明細書では今後、この通信方法を「高速通信」と呼ぶ。高速通信の例としてＶＩＡ（Compaq Computer Corp., Intel Corp., Microsoft Corp., "Virtual Interface Architecture Specification, Draft Revision 1.0," December 4, 1997, http://www.viarch.org参照）がある。高速通信とＴＣＰ／ＩＰは機能が異なるため、これらＡＰＩも異なる。

【０００５】

図２に高速通信を使うホストのソフトウェア構成例を示す。アプリケーション

１０３のプログラム１０４が高速通信ＡＰＩ９１を用いて、高速通信ライブラリ１３０を呼び出し、データ１０５を送受信する。高速通信ライブラリ１３０の通信処理１３１はカーネル１２０を介さずに高速通信ハードウェア１２を起動しデータ１０５を高速通信ネットワーク１９で通信する。高速通信におけるデータ送受信では、アプリケーション１０３が送受信したいデータ１０５のアクセス権限があるかという検査、そしてアプリケーション１０３が指定した仮想アドレスを高速通信ハードウェア１２が使う物理アドレスへの変換という二つの処理が必要である。このためアプリケーション１０３が送受信する前に、高速通信ライブラリ１３０を呼び出し、送受信するデータ１０５を登録する（登録されたデータを８０７のような角丸四角形で示す）。登録処理を高速通信ライブラリの呼び出し（１３２）でカーネルが行う（１２３）ため、アクセス権限を調査し、権限があった場合にアドレス変換を行い、登録したデータをメモリ登録テーブル１３に登録することができる。高速通信ハードウェア１２がこのメモリ登録テーブル１３を用い、アクセス権限調査とアドレス変換を行う。

高速通信ＡＰＩ９１はソケットＡＰＩ９０と異なるため、ソケットＡＰＩ９０を使うアプリケーション１００が高速通信を使用するためには、アプリケーション１００を高速通信ＡＰＩ９１に向けて書き換えなければならない。この書き換えは難しいため、多くのアプリケーションが変更されず従来のソケットＡＰＩを使いつづけ、高速通信の高速性を活用できない。この問題を解決するために、図３に示す「高速ソケット」という方式を用いる。高速ソケットライブラリ１４０はアプリケーション１００のソケットＡＰＩ９０の呼び出しを受け、エミュレーション処理１４１をし、高速通信を用い通信する。このため、アプリケーションの互換性を保ちながら、高速通信の高速性を用いることができる。高速ソケットの例として、公開特許公報特開平１１－３２８１３４、Berkeley大学の方式（S. H. Rodrigues, T. E. Anderson, D. E. Culler, "High-Performance Local Area Communication With Fast Sockets," Proceedings of the USENIX '97, 1997, pp. 257-274参照）、Shahらによる方式（H. V. Shah, C. Pu, R. S. Madukkarumukumana, "High Performance Sockets and RPC over Virtual Interface (VI) Architecture", Proceedings of CANPC'99, 1999参照）、Microsoft社のWinsock D

irect ("Winsock Direct Specification", Microsoft Windows Driver Development Kit (DDK) 参照) が挙げられる。

【０００６】

アプリケーション１００のデータ１０２を登録（８００）して通信した場合、バッファ登録８００の処理オーバーヘッド（１３２，１２３）が生じる。データ長が長い場合にこのオーバーヘッド（１３２，１２３）は通信時間に比べて短いため、高速性を得られる。一方、データ長が短いとき、通信時間に比較してこのオーバーヘッドは大きく、通信性能が低下する。この問題を解決するため高速通信ライブラリ１４０は起動時に、事前割り振りバッファ１４２をアロケートし登録（８０１）する。短いデータ１０２を通信するとき、このデータ１０２を登録せず事前割り振りバッファ１４２にコピーし通信する。この場合にはコピーのオーバーヘッドが生じるが、データ長が短くこのオーバーヘッドが登録処理に比較して少ないため、高速性を得られる。事前割り振りバッファ１４２は普段送信用バッファと受信用バッファに分かれているが、図３と今後のソフトウェア構成の図ではこれらをまとめて一つのバッファ１４２として示す。

【０００７】

以上はＴＣＰ／ＩＰ通信と高速ソケットの説明であった。一般アプリケーションがＴＣＰ／ＩＰ通信（と、その結果、ソケットＡＰＩ）を用いる一方、科学技術計算アプリケーションはＭＰＩ（Message Passing Interface Forum, "MPI: A Message-Passing Interface Standard," 1995参照）のようなＡＰＩを用いる。ＭＰＩは計算機アーキテクチャ非依存のため、高速通信の上でＭＰＩをインプリメントする場合、ＭＰＩのＡＰＩの呼び出しを高速通信のＡＰＩの呼び出しにマッピングする。この機能を実現する製品としてMPI Software Technology社のＭＰＩ－Ｐｒｏが挙げられる（R. Dimitrov and A. Skjellum., "Efficient MPI for Virtual Interface (VI) Architecture," Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, Nevada, U.S.A., June 1999, Vol.6, pp. 3094-3100参照）。図４にＭＰＩの実現方法を示す。図４では、ＭＰＩを使用するアプリケーション１０６のプログラム１０７がＭＰＩ　ＡＰＩ９２を利用してデータ１０

８を通信する。ＭＰＩライブラリ１５０がエミュレーション１５１を行い、上記のマッピングを行う。ＭＰＩ（図４）の構成は高速ソケット（図３）の構成と同様のため、両者の通信における課題も同様である。本明細書では記載がなければ、高速ソケットに説明する方法はＭＰＩにも当てはまり、またＭＰＩに説明する方法は高速ソケットにも当てはまる。

【０００８】

【発明が解決しようとする課題】

本発明は従来の高速ソケットライブラリ１４０やＭＰＩライブラリ１５０のような通信ライブラリの５つの問題を解決する（下記にこれらのライブラリを「エミュレーションライブラリ」と呼ぶ）。ここではこれらの問題を概説して、必要な場合に発明の実施の形態の説明ではこれらの問題の詳細な説明をしてから本発明の解決手段を説明する。

第一の問題は次のとおりである。従来方式では送信ホストがデータ長により、送信ホストにデータ１０２，１０８を登録（８００，８０８）した通信と事前割り振りバッファ１４２，１５２にコピーした通信のどちらが最適かを選択するが、受信ホストにどちらが最適かを考慮しない。このため、受信ホストの受信処理性能を低下する。

【０００９】

第二の問題は次のとおりである。受信ホストで受信呼び出しが受信データよりを先行した場合、受信ホストが受信データ１０２，１０８領域を登録（８００）してこのアドレスとデータ長を通信相手に知らせることができる。しかし、送信ホストが送信開始後にこの知らせを受信した場合、この知らせは無駄となり、送信ホストと受信ホストの処理オーバーヘッドとなり、ネットワークバンド幅を占めるため、システム全体の処理性能を低下する。

【００１０】

第三の問題は次のとおりである。従来方式は送信ホストからのデータ書き込みと受信ホストからのデータ読み出しという二つのデータ転送方法と、受信ホストと送信ホストそれぞれのデータ１０２，１０８を登録（８００，８０８）した通信と事前割り振りバッファ１４２，１５２にコピーした通信の４つの組み合わせ

、全体で８つの組み合わせを全て利用することができない。このため、高速通信を可能にするネットワークの性能を最大限に向上できない。

【００１１】

第四の問題は次のとおりである。従来方式は通信相手にもかかわらず同じ通信方法を使用する。しかし、今後は通信相手がサーバ等のコンピュータでなく、ｉＳＣＳＩ（ＴＣＰ／ＩＰ上ＳＣＳＩプロトコル、J. Satran et alli., "iSCSI (Internet SCSI)," Internet Engineering Task Force Internet-Draft draft-satran-iscsi-01.txt, July 10, 2000参照）を使用しているストレージ装置であることが考えられる（本発明では、通信する装置を種類にかかわらず「ホスト」と呼ぶ）。ストレージ装置はコンピュータに比較して事前割り振りバッファ１４２に使用できるメモリ量が制限されており処理性能が低いことがあるため、上記第三の問題に述べた８つの組み合わせの一部のみが効率的である。通信相手の特性により通信方法を制限しないことは、例えばこの通信相手がストレージ装置の場合には装置の必要となるメモリなどを増加し、送受信処理を複雑にし装置の必要な処理能力を高め、コストを高くする。

【００１２】

第五の問題は次のとおりである。従来方式はＴＣＰ／ＩＰ接続確立時に事前割り振りバッファ１４２，１５２をアロケートし、この後の通信にはバッファの大きさ等を変更しない。このため、このＴＣＰ／ＩＰ接続の特性に必要となるバッファ量を適応することができない。例えば必要な時にバッファの大きさを増加しないことは性能を低下する要因になる。そして、事前割り振りバッファ１４２，１５２のような登録（８０１，８０９）したデータ領域は、データ送受信対象のためスワップアウトできなく、主記憶を占める。このため、バッファの大きさを削減しないことは他のアプリケーションが使えるメモリを少なくするため性能低下の要因にもなる。

【００１３】

【課題を解決するための手段】

第一の問題の解決方法は、通信するホストが通信相手にデータ１０２，１０８を登録（８００，８０８）した通信と事前割り振りバッファ１４２，１５２にコ

ピーした通信のどれが最適かを決定するデータ長を知らせることである。

【００１４】

第二の問題の解決方法は、受信ホストが知らせの効果を計算し、効果が低い場合に知らせを抑えることである。

【００１５】

第三の問題の解決方法は、８つの組み合わせを可能にする通信プロトコルである。

【００１６】

第四の問題の解決方法は、送受信動作に期待される転送データ長を通信相手に知らせることである。

【００１７】

第五の問題の解決方法は通信パターンによるバッファの変更である。

【００１８】

【発明の実施の形態】

＜＜第一の問題の解決方法＞＞

この問題の解決方法の説明としてまず、従来方式を説明する。図５にＭＰＩ－Ｐｒｏの通信方法を示す。（今後、通信方法の図を理解しやすくするために、図３と図４のアプリケーション１００，１０６とエミュレーションライブラリ１４０，１５０のみを示す。両ホスト１０，２０は同様なソフトウェア構成を持つ。そして、片方向のデータ転送のみを示し、左のホストを送信ホスト１０、右のホストを受信ホスト２０とする。）ＭＰＩ－Ｐｒｏは送信側では事前割り振りバッファを利用しなく、アプリケーション１０６のデータ１０８から直接送信する。全ての通信は送信ホスト１０からの書き込みである。データ長が長い場合にデータ１０８を直接アプリケーション２０６データ２０８に送信（９００）し、データ長が短い場合データを受信ホスト２０の事前割り振りバッファ２５２に送信（９０２）する。ここでは、どちらに送信するか決定するホストは送信ホスト１０である。

【００１９】

スーパーコンピュータの場合、ホスト１０、２０は普段全て同じ物であるため

、送信ホスト１０は受信ホスト２０のアプリケーションデータ２０８と事前割り振りバッファ２５２とのどちらに送信すれば最適かを判断できる。しかし、高速ソケット通信やＭＰＩを実行するワークステーションクラスタのようにホスト１０，２０が異なるシステムの場合、ホストによりメモリ登録動作（１３２，１２３）の時間とメモリコピーの性能が異なるため、送信ホスト１０だけの判断は不可能である。判断を間違えば受信処理（と、その結果、送信ホスト１０と受信ホスト２０を含むシステム全体）の性能が低下する。

【００２０】

以上は従来の技術である。本発明ではこの問題を解決するために、受信ホストが登録（８０５）した通信と事前割り振りバッファ２５２を介した通信のどれが最適かを決定するデータ長を送信ホストに知らせる。知らせるタイミングはまず、高速ソケットでは通信するホスト１０，２０がソケットＡＰＩ９０でソケットの接続を確立したとき、ＭＰＩではＭＰＩライブラリ１５０，２５０の初期化時である（今後、このタイミングを「通信開始」と呼ぶ）。従来（図６ａ）このタイミングで送信するデータ９１０（事前割り振りバッファアドレスとデータ長等）と一緒に、本発明のデータ長の知らせ９１１（図６ｂ）を転送することが考えられる。そしてもう一つの可能なタイミングとして、ホスト２０が始めてホスト１０に通信したとき、この情報を追加することも考えられる。

【００２１】

どちらの通信方法が最適かを決定するデータ長の設定として、（１）アプリケーション２０６からの設定、（２）ホスト１０，２０の管理者やユーザやアプリケーションからの設定、（３）エミュレーションライブラリ１４０，１５０をホスト１０，２０にインストールしたプログラムの設定、などの方法が考えられる（しかし、これらの方法に限られていない）。

【００２２】

以上の発明のため、受信ホスト２０の受信処理（と、その結果、システム全体）の性能が向上する、という効果を得る。

【００２３】

＜＜第ニの問題の解決方法＞＞

この問題の解決方法の説明としてまず、従来方式を説明する。図７に従来方式を示す。受信ホスト２０のアプリケーション２０６が受信呼び出しを実行し、エミュレーションライブラリ２５０が、アプリケーションデータ２０８に直接受信することが効率的であることを判断したとき、データ２０８を登録（８０５）して、送信側に受信アドレスとデータ長を知らせること（９５０）ができる（データ転送以外、エミュレーションライブラリ１４０，１５０，２５０は制御メッセージを交換し、このアドレスとデータ長の知らせを制御メッセージとして転送する）。この場合、送信ホスト１０が送信呼び出しを実行したときにデータをこのアドレスに送信して（９５１）、そして送信完了の確認メッセージ９５２を送信する。このため、送信呼び出しの直後に送信の開始ができる。しかし、以前述べたとおり、送信ホスト１０が送信開始後にアドレスの知らせ９５０を受信した場合、この知らせ９５０は無駄となり、処理オーバーヘッドとなり、ネットワークバンド幅を占めるため、システム全体の処理性能を低下する。

【００２４】

以上は従来の技術である。本発明はこの問題を解決するために、受信ホスト２０がアドレスの知らせ９５０の効果を計算し、効果が低い場合に知らせを抑える。送信したアドレスの知らせ９５０の送信回数に対して、このアドレスに受信した回数の割合で効果を計算できる。そして、この効果があるしきい値より低い場合、アドレスの知らせ９５０の送信を抑える。

【００２５】

上記の解決方法にはまず、ユーザや管理者、エミュレーションライブラリ１４０，１５０，２５０作者かインストールプログラム、あるいはアプリケーション２００がしきい値を設定することが考えられる。そして、全てのアドレスの知らせ９５０をまとめて効果を計算すること、そして受信アドレス毎に計算すること、の２つの方式が考えられる（後者の場合、効率の悪い受信アドレスだけに、アドレスの知らせ９５０を抑えることができる）。そして、抑える動作として中止（止めて続けない）と中断（止めた後に続く）が考えられる。

【００２６】

以上の発明のため、送信ホスト１０と受信ホスト２０の処理効率を向上し、ネ

ットワークバンド幅を無駄に占めないため、これらのホスト（と、その結果、システム全体）の性能が向上する、という効果を得る。

【００２７】

＜＜第三の問題の解決方法＞＞

ここではまず、従来方式の通信方法を説明する。今後送信個所と受信個所の組み合わせを示す番号（９００，９０４等）に、送信ホスト１０からの書き込み（ｗｒｉｔｅ）か受信ホスト２０からの読み出し（ｒｅａｄ）を加えて各組み合わせを示す。例えば、以前説明した図５のＭＰＩ－Ｐｒｏは９００－ｗｒｉｔｅと９０４－ｗｒｉｔｅの２つの組み合わせのみを使用する。

【００２８】

図８にWinsock Directの通信方法を示し、図９にプロトコルの詳細を示す。Winsock Directではまず、送信ホスト１０がデータを事前割振りバッファ１４２，２４２の間でデータ送信する（９４０，９３０）（９０３－ｗｒｉｔｅ）。受信ホスト２０が受信したデータをアプリケーション２００のデータ２０２にコピーする（９０５，９３１、９４２）。データ長が長い場合、上記で先頭データのみを送信し、残りのデータ１０２を登録し（８００）、その先頭アドレスを上記の送信９４０，９３０に加える。受信ホストがデータ２０２を登録（８０２）する。高速通信ハードウェア１２が受信ホスト２０からの読み出し通信の機能がある場合、受信ホスト２０が通信データを読み出す（９３２，９００－ｒｅａｄ）。一方、受信側からの読み出し通信機能がない場合受信ホストが受信領域の先頭アドレスを知らせ（９４１）、送信ホスト１０がデータを書き込む（９４３，９００－ｗｒｉｔｅ）。この後、最後に通信をしたホストが通信完了の確認を送信する（９３３，９４４）。そして、両ホスト１０，２０がメモリ登録（８００，８０２）を解除する。

【００２９】

図１０にShahらによる方式の通信方法を示す。送信ホスト１０はデータ長が短い場合、事前割振りバッファ１４２，２４２間でデータを送信する（９０３－ｗｒｉｔｅ）。一方データ長が長い場合データ１０２を登録（８００）して、受信ホストの事前割り振りバッファ２４２に送信する（９０４－ｗｒｉｔｅ）。

【００３０】

以上は従来方式である。本発明は、図１１に示すとおり、８つの組み合わせを全て利用可能にするプロトコルを使用する。特にこのプロトコルは従来方式が利用しなかった９０２－ｒｅａｄ、９０２－ｗｒｉｔｅ、９０３－ｒｅａｄ、９０４ｒｅａｄを可能にする。

【００３１】

以下に、本発明の通信方法を説明する。図１２に送信ホスト１０側のアルゴリズムを示す。まず、受信したアドレス知らせメッセージがあれば、これらのメッセージを処理する（７０１）。そして送信データ１０２，１０８のデータ長を調べ（７０２）、データが長い場合にメモリを登録（８００，８０８）し（７０４）、短い場合に事前割り振りバッファ１４２，１５２にコピーする（７０３）。

【００３２】

次に、アドレス知らせメッセージで知らせた、受信ホスト２０での宛先アドレスがあれば（７０５）、送信データを受信ホスト２０のアプリケーションデータ２０２、２０８に書き込み送信する（７０６）（長いデータ長の場合９００－ｗｒｉｔｅ、短いデータ長の場合９０２－ｗｒｉｔｅになる）。宛先アドレスがなければ、受信ホスト２０の事前割り振りバッファ２４２，２５２への送信が可能か（すなわち、事前割り振りバッファに空きがあるか）、そして適切か（第一の問題で説明したとおり、受信ホスト２０がこのデータ長を事前割り振りバッファ４２，２５２で受信したいか）を調べる（７０７）。この二つの条件が真であれば、送信ホスト１０が事前割り振りバッファ２４２、２５２に書き込み送信する（７０８）（長いデータ長の場合９０４－ｗｒｉｔｅ、短いデータ長の場合９０３－ｗｒｉｔｅになる）。一方、この二つの条件のどれかが真でなければ、送信データのアドレス知らせを送信して（７０９）、受信完了メッセージを待つ（７１０）（長いデータ長の場合９００－ｒｅａｄか９０４－ｒｅａｄのどれか、短いデータ長の場合９０２－ｒｅａｄか９０３－ｒｅａｄのどれかになる）。最後に、送信データを解放（７１１）する（長いデータ長の場合登録８００，８０８を、短いデータ長の場合事前割り振りバッファ１４２，１５２を解放する）。

【００３３】

図１３に受信側のアルゴリズムを示す。まず、事前割り振りバッファ２４２，２５２で受信したデータをコピー（９０５）して、アドレス知らせメッセージがあるかを調べる（７２１）。アドレス知らせメッセージがあった場合（７２２）、データ長を調べる（７２３）。データ長が長い場合、アプリケーションデータ２０２，２０８を登録（８０２，８０５）し（７２４）、送信ホスト１０からデータを読み出す（７２５）（９００－ｒｅａｄか９０２－ｒｅａｄのどれかになる）。一方、データ長が短い場合、受信ホスト２０が事前割り振りバッファ２４２，２５２にデータを読み出す（７２６）（９０３－ｒｅａｄか９０４－ｒｅａｄのどれかになる）。データ長にもかかわらず、最後に受信完了メッセージを送信する（７２７）。

【００３４】

アドレス知らせメッセージがなかった場合（７２２）、データ長を調べる（７２８）。データ長が短い場合、事前割り振りバッファ２４２，２５２でのデータ受信（９０３－ｗｒｉｔｅか９０４－ｗｒｉｔｅ）か、アドレス知らせメッセージを待つ（後者の場合、図１３の処理をスタート７２０から繰り返す）。一方、データ長が長い場合にはアプリケーションのデータを登録して（７２９）、この先頭アドレスをアドレス知らせメッセージで送信する（７３０）。送信ホスト１０では送信処理開始の前にこのアドレス知らせメッセージが受信されたら、９００－ｗｒｉｔｅと９０２－ｗｒｉｔｅのどれかの通信になる。一方、受信ホスト２０がこのステップでアドレス知らせメッセージを受信すれば、これは送信ホスト１０と受信ホスト２０が同時にお互いにアドレス知らせメッセージを送信したことが分かる。この場合、送信ホスト１０に送信してもらうために、受信ホスト２０がこのデータ転送におけるアドレス知らせメッセージを無視する。

【００３５】

以上の発明のため、送信ホスト１０と受信ホスト２０の間の通信性能が向上し、これらのホスト（と、その結果、システム全体）の性能が向上する、という効果を得る。

【００３６】

＜＜第四の問題の解決方法＞＞

ストレージ装置などのホスト１０，２０はアプリケーションデータ１０２，２０２，１０８，２０８か通信割り振りバッファ１４２，１５２，２４２，２５２のどれかしか装備しないことが考えられる。第三の問題の解決方法で説明した通信アルゴリズムはこの場合にでも使用できる。あるホスト１０，２０にアプリケーションデータ１０２，１０８，２０２，２０８がない場合、このホスト１０，２０の処理の判断７０２，７２３，７２８をいつも「短い」とする。逆にあるホスト１０，２０に事前割り振りバッファ１４２，２４２，１５２，２５２がなければ、このホストでこれらの判断をいつも「長い」とし、そして通信開始にこのホストから図６ａの事前割り振りバッファアドレスを送信しなく、そして通信相手に判断７０７の「可能かつ適切か」の条件に「存在するか」という条件を加える。このため、必要でない機能のインプリメントが不要となり、そして事前割り振りバッファ１４２，２４２，１５２，２５２がない場合このメモリ領域のアロケーションが不要となり、このアルゴリズムは容易なインプリメントと資源の節約を可能にする。しかし、下記に説明する問題が生じる。

【００３７】

上記のアルゴリズムを使用しホストとストレージ装置が通信している場合、ストレージ装置は必要でない資源（事前割り振りバッファ１４２，２４２，１５２，２５２等）をアロケートしない。一方、ホスト側は通信の特性を理解しないため、例えばデータ転送単位がいつも長い時にでも事前割り振りバッファ１４２，２４２，１５２，２５２をアロケートし、メモリを無駄にする。

【００３８】

本発明では上記の問題を解決するために通信初期化時に期待される転送データ長を使用してライブラリの初期化を行う。この転送データ長を通信相手に知らせ、およびまたはアプリケーション１００，２００，１０６，２０６が指定する。この転送データ長が「長い」か「短い」により、アプリケーションのデータ送受信が必要か、または事前割り振りバッファ１４２，２４２，１５２，２５２が必要かを判断できる。

【００３９】

以上の発明のため、ホスト１０，２０の間の通信性能が向上し、メモリを節約

するため、これらのホスト（と、その結果、システム全体）の性能が向上する、という効果を得る。そしてホスト１０，２０に必要な処理性能とメモリ量だけを装備すればよいため、システムのコストを低下できる、という効果もある。

【００４０】

＜＜第五の問題の解決方法＞＞

次に本発明の解決方法を説明する。まず、事前割り振りバッファの変更は（１）拡大か縮小のサイズ変更、（２）追加か削除、（３）受信用バッファを送信用にすることか、送信用バッファを受信用にすること、の３種類がある。

【００４１】

ホスト１０，２０は次の動作で変更を決定することが考えられる。まず、エミュレーションライブラリ１４０，１５０，２４０，２５０の起動時に、サイズの最大値と最小値、そして使用率の上限と下限の値を設定する。これらの値の設定方法は（１）ライブラリ１４０，１５０作成時の定数（２）ホスト１０，２０のユーザや管理者やユーザやアプリケーションからの設定、（３）ライブラリ１４０、１５０，２４０，２５０をホスト１０，２０にインストールしたプログラムの設定、などの方法が考えられる（しかし、これらの方法に限られていない）。そして、通信開始後、送受信動作毎およびまたは定期的に送信用事前割り振りバッファ１４２，１５２と受信用事前割り振りバッファ２４２，２５２の使用率を調べ、平均使用率を計算する。この平均使用率が上限を超え、そしてこの事前割り振りバッファ１４２，２４２，１５２，２５２のサイズが最大限を超えていない場合，バッファの拡大や追加を行う。逆に、この平均使用率が下限を超え、そしてこの事前割り振りバッファ１４２，２４２，１５２，２５２のサイズが最小限を超えていない場合、バッファの縮小や削除を行う。そして送信用バッファにある変更、そして受信用バッファにその逆の変更を決定したら、バッファの用途を変更する（逆もまた同様である）。例えば、送信用事前割り振りバッファ１４２，１５２を拡大して受信用事前割り振りバッファ２４２，２５２を縮小する場合、受信用バッファの一部を送信用にすることが考えられる。

【００４２】

受信ホスト２０での事前割り振りバッファ２４２，２５２を変更した場合、受

信ホスト２０が送信ホスト１０に変更内容を制御メッセージで知らせる必要がある（逆に、送信ホスト１０の送信用事前割り振りバッファ１４２，１５２の変更を受信ホスト２０に知らせる必要はない）。サイズ縮小、バッファ削除と用途変更の変更知らせメッセージの場合、送信ホストが変更される領域にデータを送信しないために、受信ホスト２０が変更知らせメッセージを送信して、送信ホストが応答した後に変更を行う。これら以外の変更を、知らせメッセージを行う前にでも変更が行えられ、そして送信ホストの応答が不要である。

【００４３】

以上の発明のため、ホスト１０，２０の間の通信性能が向上し、メモリを節約するため、これらのホスト（と、その結果、システム全体）の性能が向上する、という効果を得る。そしてホスト１０，２０に必要なメモリ量だけを装備すればよいため、システムのコストを低下できる、という効果もある。

【００４４】

＜＜変形例＞＞

本発明はすでに記載した実施の形態あるいはその変形例に限定されるのではなく、以下に例示する変形例あるいは他の変形例によっても実現可能であることは言うまでもない。また、上記複数の実施の形態あるいはその変形例として記載の技術あるいは以下の変形例の組み合わせによっても実現できる。

（１）以上の説明ではデータ１０２，２０２，１０８，２０８を登録（８００，８０２，８０５，８０６）して通信した場合、通信完了後に登録を解除すると述べている。しかし、ＭＰＩ－Ｐｒｏと同様に、後で同じアドレスのデータが通信された場合に登録を不要にするために登録を解除しなくキャッシングすることが考えられる。

（２）以上のアルゴリズムやプロトコルの説明では通信完了確認メッセージの送信を示したが、高速通信ハードウェア１２や通信プロトコルの機能によりこれらのメッセージ、あるいはその一部が不要となる。

（３）上記の５つの問題の解決方法を別々に使用すること、あるいは複数同時に組み合わせて使用することができる。

【００４５】

なお、本発明を実施するためのプログラムは、それ単独であるいは他のプログラムと組み合わせて、ディスク記憶装置等のプログラム記憶媒体に記憶された販売することができる。また、本発明を実施するためのプログラムは、すでに使用されている通信を行うプログラムに追加される形式のプログラムでもよく、あるいはその通信用のプログラムの一部を置換する形式のプログラムでもよい。

【００４６】

【発明の効果】

以上から明らかなように、通信を高速化し、処理オーバーヘッドとメモリ使用量を減らすことができる。

【図面の簡単な説明】

【図１】

ＴＣＰ／ＩＰプロトコルを使用し通信するホストのソフトウェア構成を示す図。

【図２】

高速通信を使用し通信するホストのソフトウェア構成を示す図。

【図３】

高速ソケットを使用し通信するホストのソフトウェア構成を示す図。

【図４】

ＭＰＩを使用し通信するホストのソフトウェア構成を示す図。

【図５】

ＭＰＩ－Ｐｒｏの通信方法を示す図。

【図６】

第一の問題を解決するための、通信方法切り替えしきい値のデータ長の転送を示す図。

【図７】

送信宛先を知らせるためのアドレス知らせメッセージとその応答を示す図。

【図８】

Winsock Directの通信方法を示す図。

【図９】

Winsock Directのプロトコルの詳細を示す図。

【図１０】

Shahらによる方式の通信方法を示す図。

【図１１】

本発明の通信方法を示す図。

【図１２】

本発明の送信側の通信アルゴリズムを示す図。

【図１３】

本発明の受信側の通信アルゴリズムを示す図。

【符号の説明】

１０：送信ホスト

２０：受信ホスト

１００，１０３，１０６，２００：アプリケーション

１２０：オペレーティング・システム・カーネル

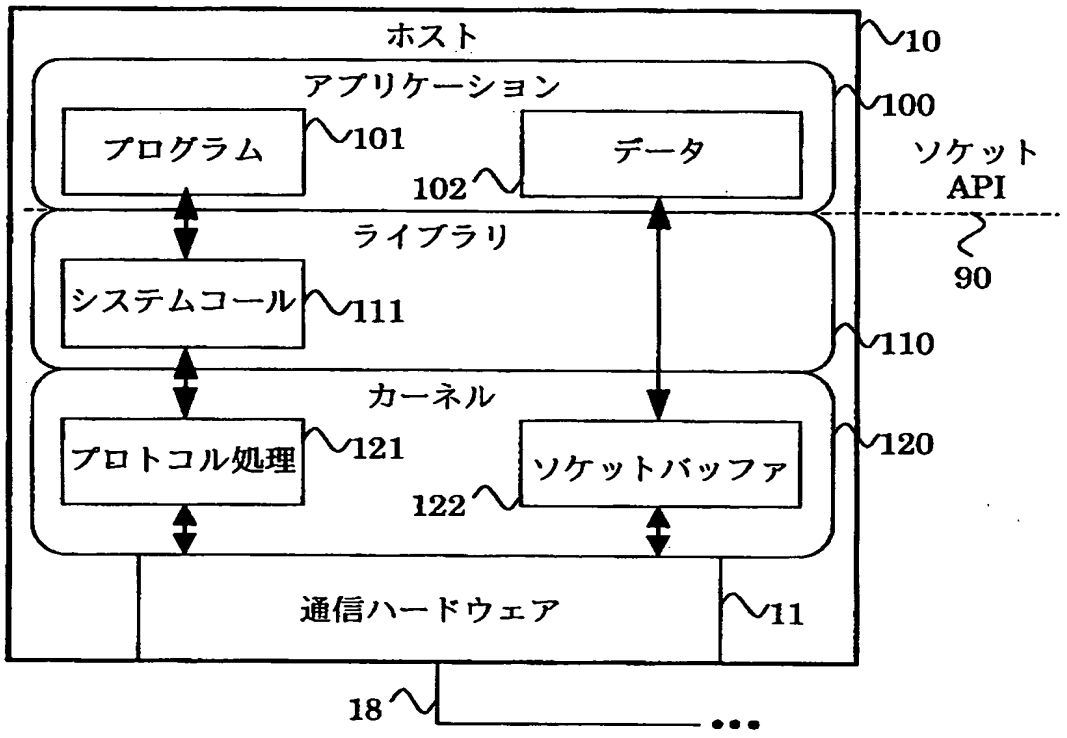
１１：通信ハードウェア

１２：高速通信ハードウェア。

【書類名】　図面

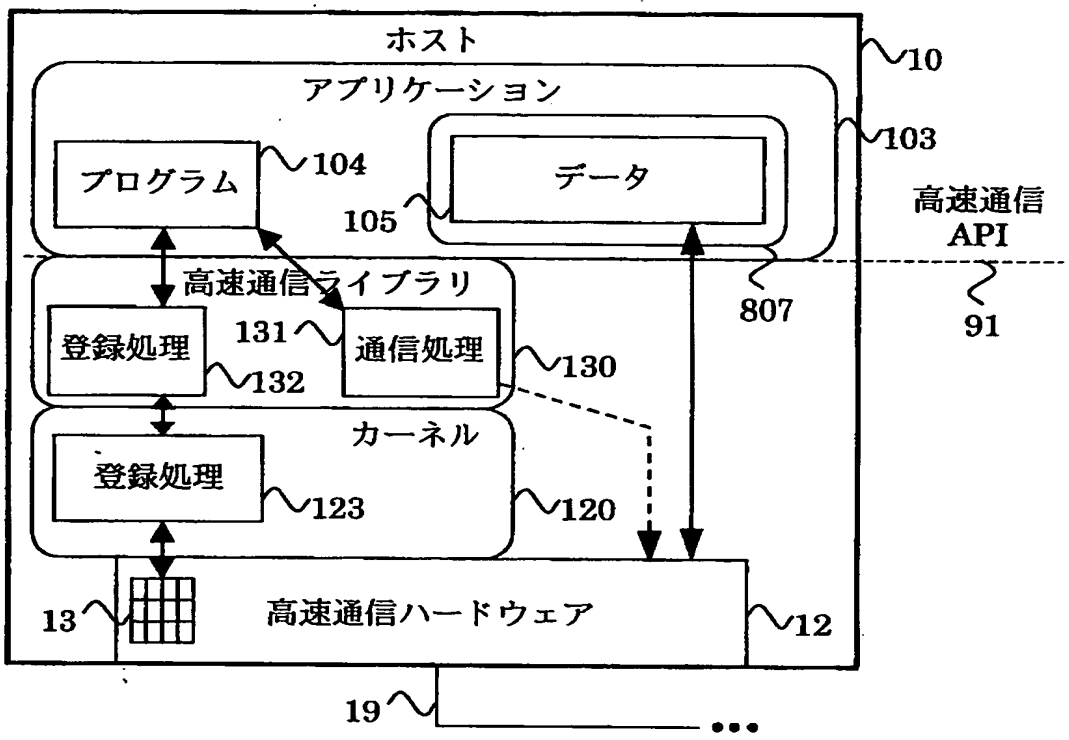
【図１】

図１


ホスト
10
アプリケーション
100
プログラム
101
データ
102
ソケット
API
90
ライブラリ
システムコール
111
110
カーネル
プロトコル処理
121
ソケットバッファ
122
120
通信ハードウェア
11
18

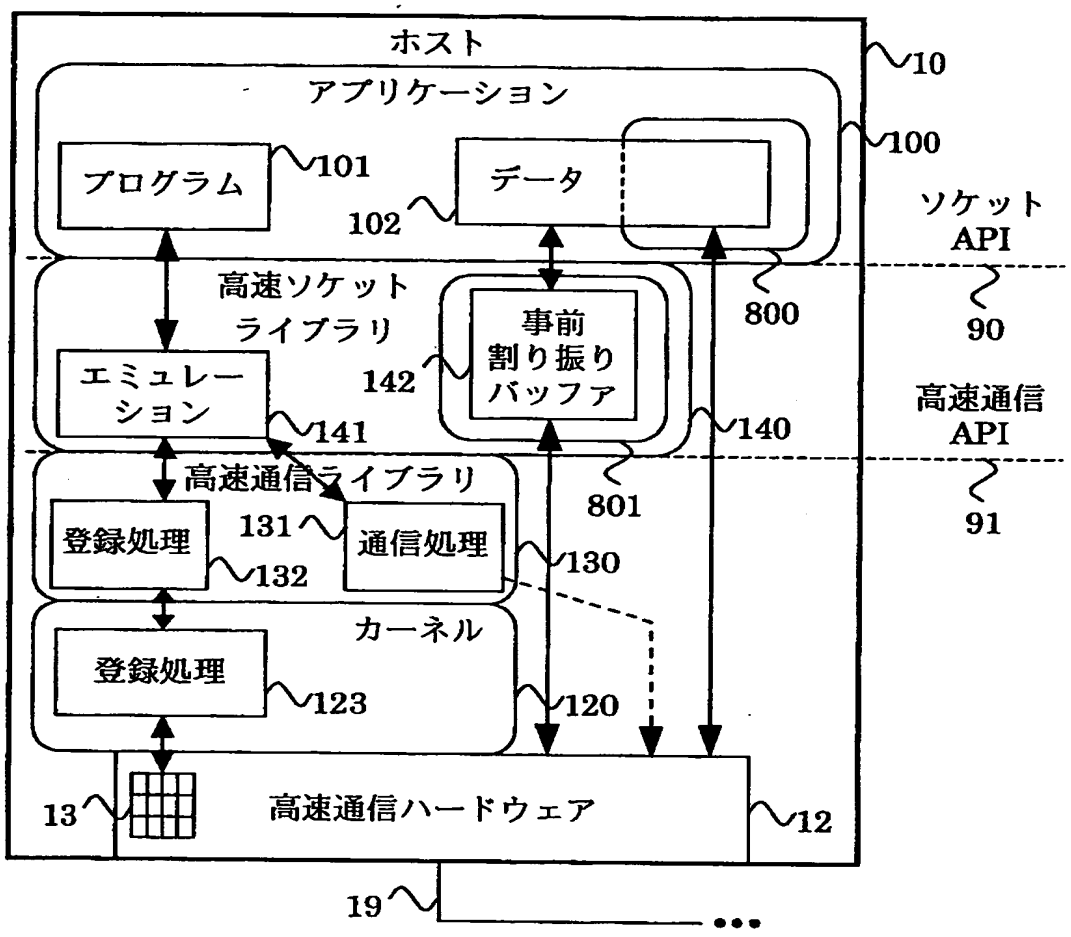
【図２】

図2


ホスト
10
アプリケーション
103
プログラム
104
データ
105
高速通信
API
高速通信ライブラリ
807
91
登録処理
131
通信処理
130
132
カーネル
登録処理
123
120
13
高速通信ハードウェア
12
19

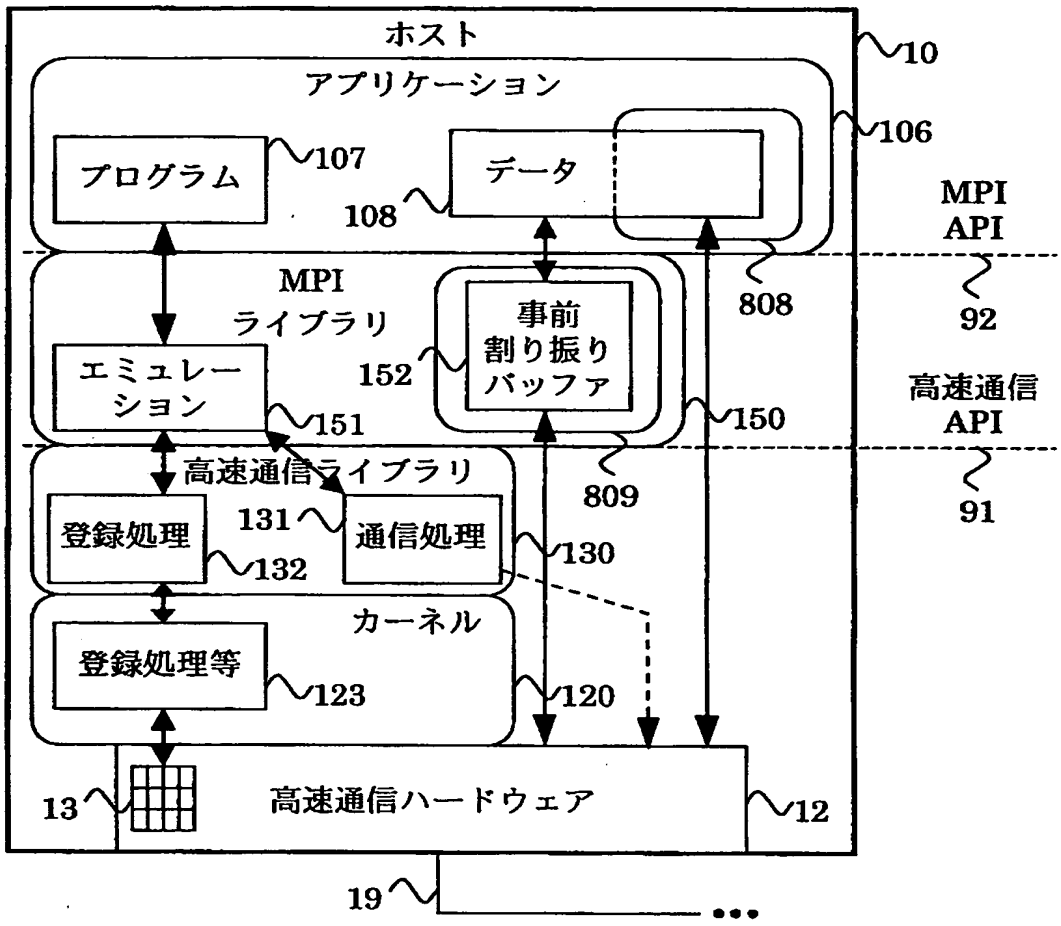
【図３】

図３


ホスト
10
アプリケーション
100
プログラム
101
データ
102
ソケット
API
90
高速ソケット
ライブラリ
800
事前
割り振り
バッファ
142
エミュレー
ション
141
140
高速通信
API
91
801
高速通信ライブラリ
登録処理
131
通信処理
132
130
カーネル
登録処理
123
120
13
高速通信ハードウェア
12
19

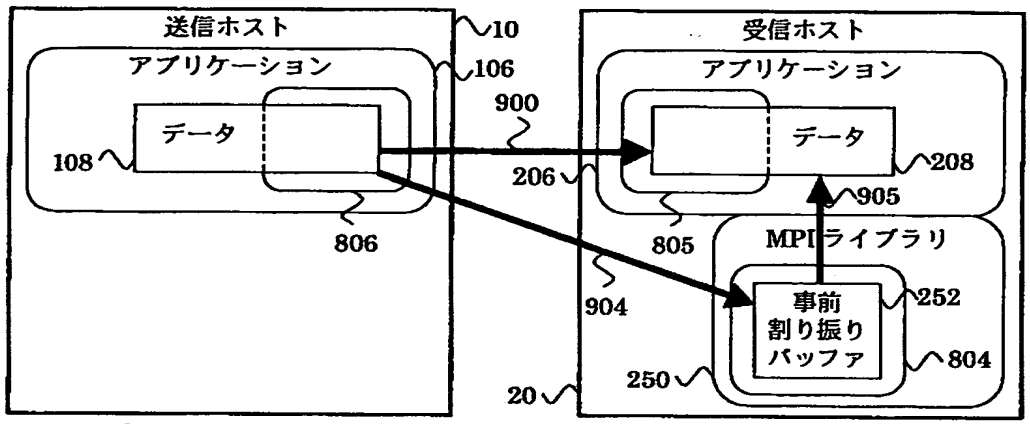
【図４】

図４


ホスト
10
アプリケーション
106
プログラム
107
データ
108
MPI
API
92
808
MPI
ライブラリ
事前
割り振り
バッファ
152
エミュレー
ション
151
150
高速通信
API
91
809
高速通信ライブラリ
登録処理
131
通信処理
130
132
カーネル
登録処理等
123
120
13
高速通信ハードウェア
12
19

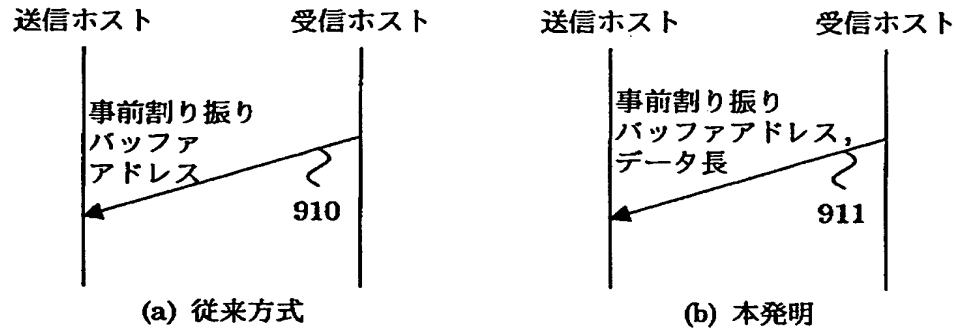
【図５】

図５


送信ホスト
10
アプリケーション
106
データ
108
900
806
受信ホスト
アプリケーション
データ
208
206
905
805
MPIライブラリ
904
事前
割り振り
バッファ
252
804
250
20

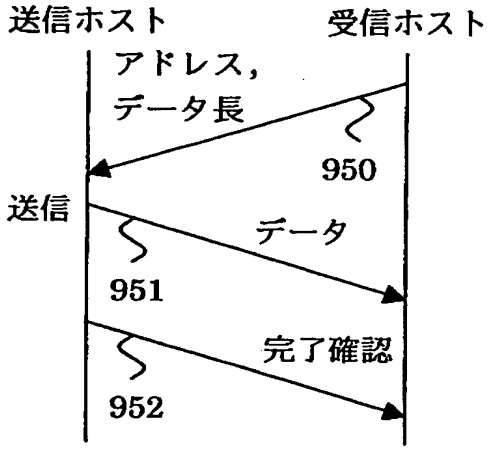
【図６】

図６


送信ホスト
受信ホスト
事前割り振り
バッファ
アドレス
910
(a) 従来方式
送信ホスト
受信ホスト
事前割り振り
バッファアドレス，
データ長
911
(b) 本発明

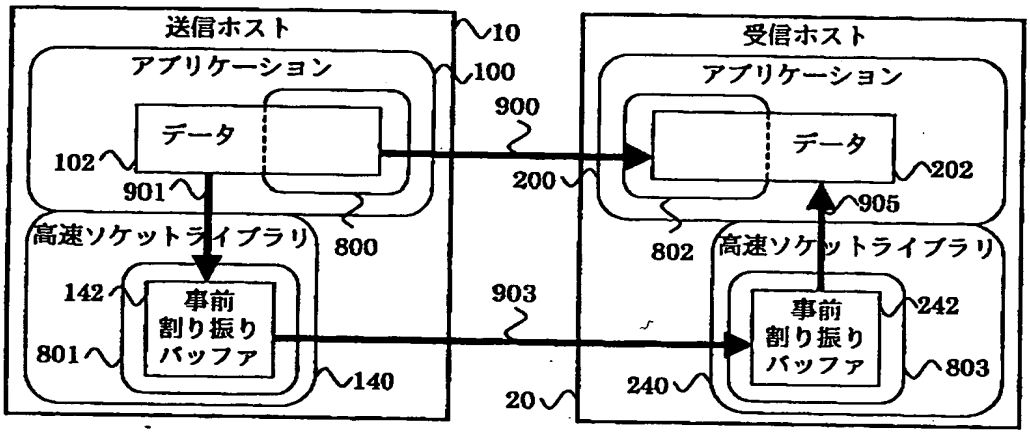
【図７】

図７


送信ホスト
受信ホスト
アドレス，
データ長
950
送信
データ
951
完了確認
952

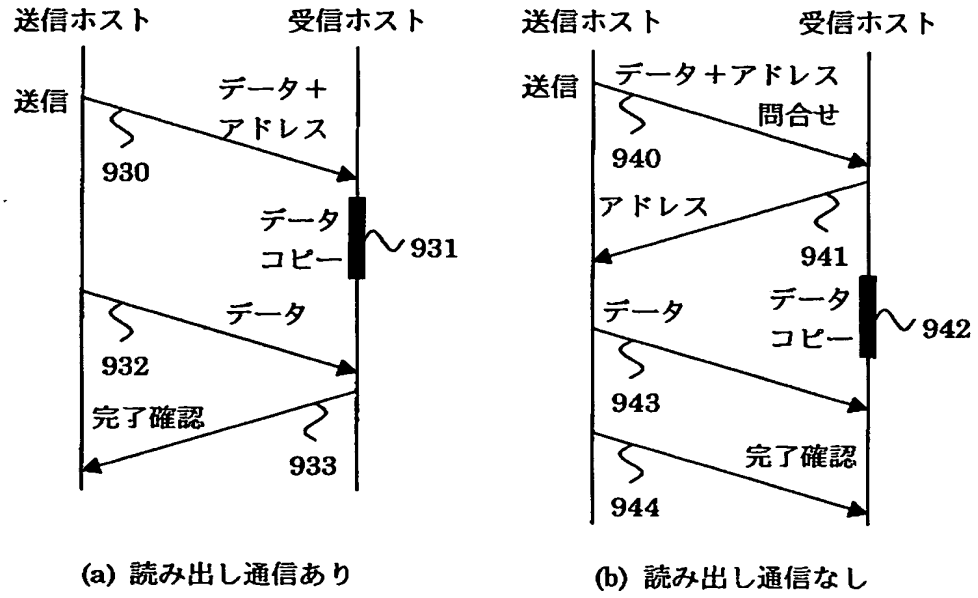
【図８】

図８


送信ホスト
10
アプリケーション
100
データ
102
901
高速ソケットライブラリ
800
142
事前
割り振り
バッファ
801
140
900
903
20
受信ホスト
アプリケーション
データ
200
202
905
802
高速ソケットライブラリ
242
事前
割り振り
バッファ
240
803

【図９】

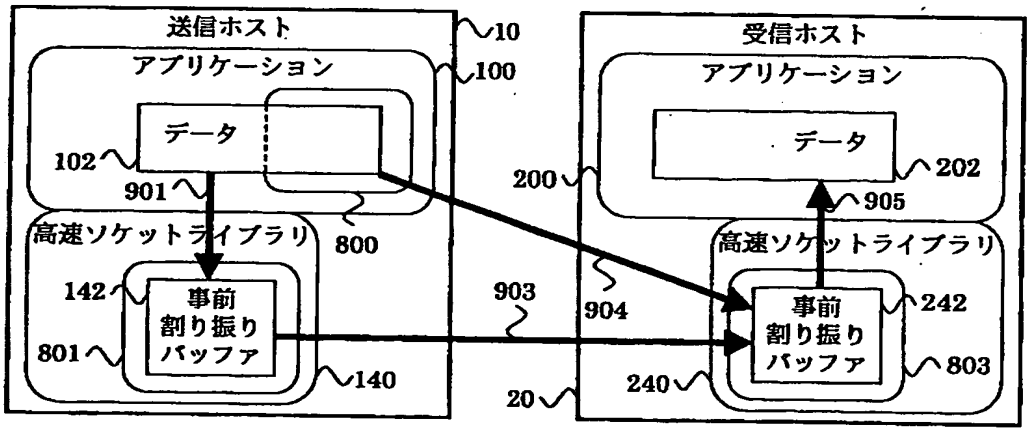
図９



(a) 読み出し通信あり　　(b) 読み出し通信なし

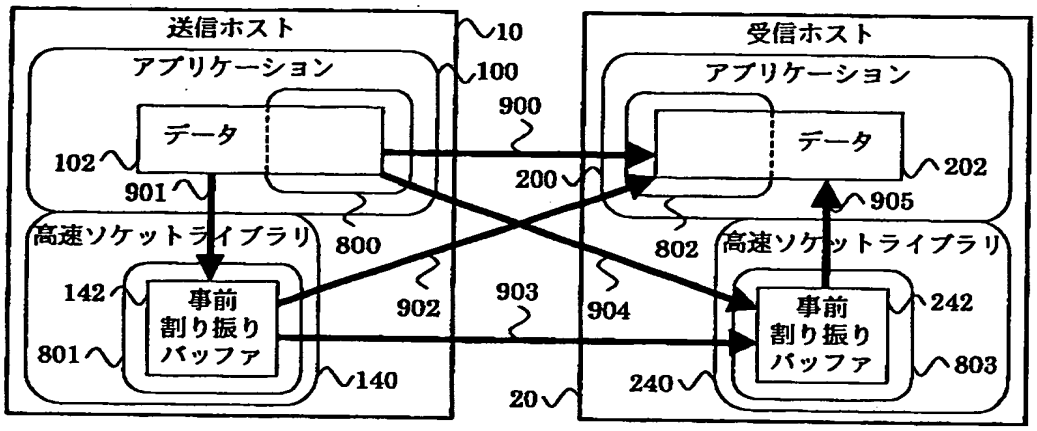
【図１０】

図１０



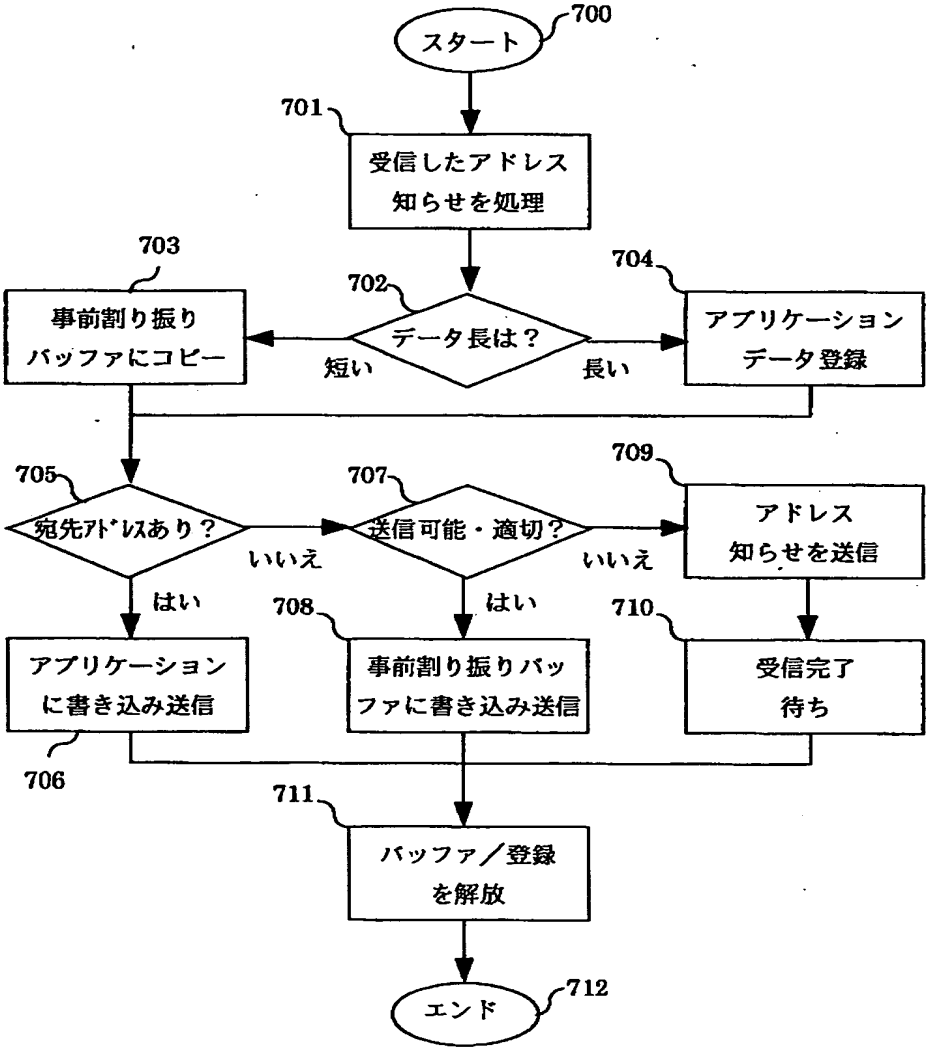
【図１１】

図１１


送信ホスト
10
アプリケーション
100
900
データ
102
901
200
データ
202
905
高速ソケットライブラリ
800
802
高速ソケットライブラリ
142
事前
割り振り
バッファ
902
903
904
事前
割り振り
バッファ
242
801
140
240
803
20

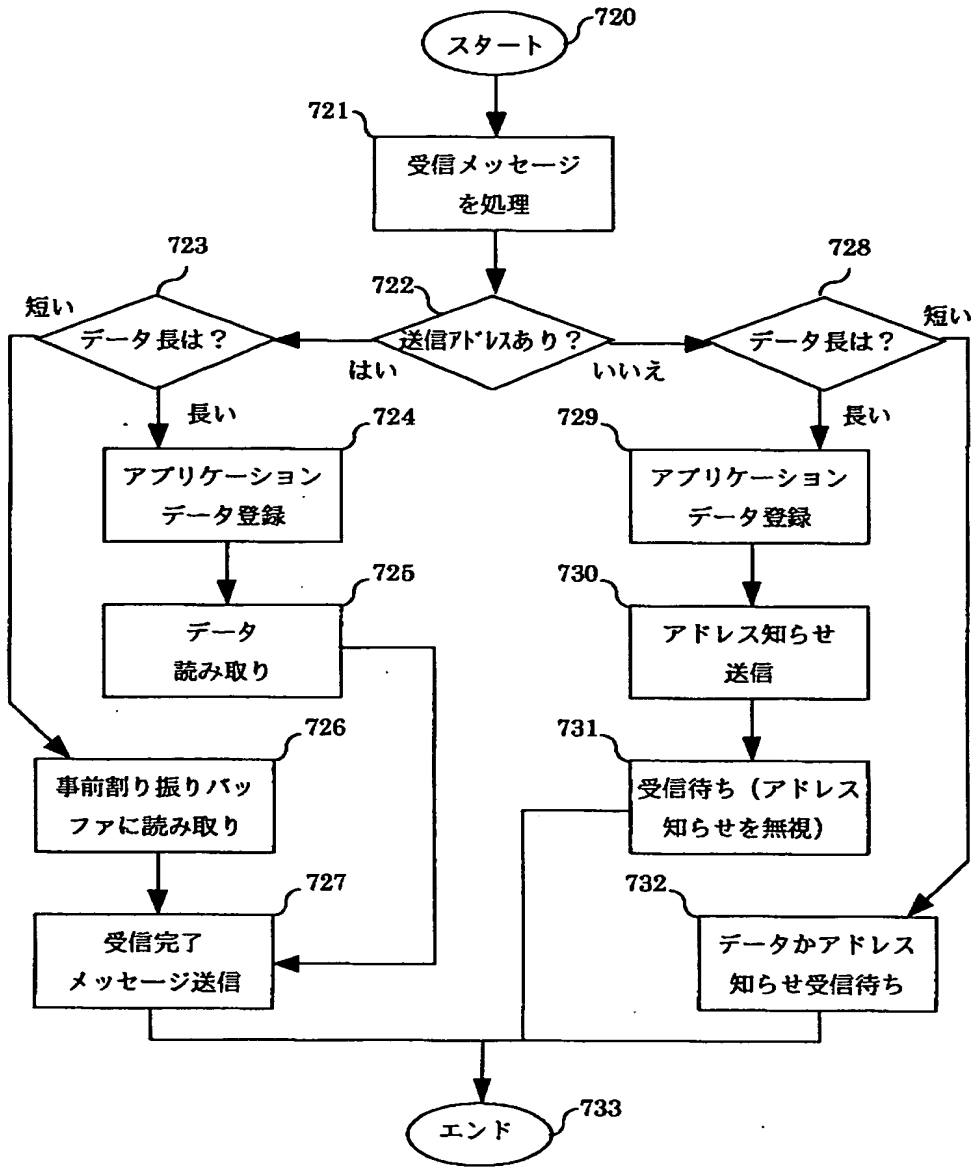
【図１２】

図１２


スタート
700
701
受信したアドレス
知らせを処理
702
データ長は？
短い
長い
703
事前割り振り
バッファにコピー
704
アプリケーション
データ登録
705
宛先ｱﾄﾞﾚｽあり？
いいえ
はい
707
送信可能・適切？
いいえ
はい
709
アドレス
知らせを送信
708
事前割り振りバッ
ファに書き込み送信
710
受信完了
待ち
アプリケーション
に書き込み送信
706
711
バッファ／登録
を解放
エンド
712

【図１３】

図１３


スタート
720
受信メッセージ
を処理
721
送信ｱﾄﾞﾚｽあり？
722
はい
いいえ
データ長は？
723
短い
長い
アプリケーション
データ登録
724
データ
読み取り
725
事前割り振りバッ
ファに読み取り
726
受信完了
メッセージ送信
727
データ長は？
728
短い
長い
アプリケーション
データ登録
729
アドレス知らせ
送信
730
受信待ち（アドレス
知らせを無視）
731
データかアドレス
知らせ受信待ち
732
エンド
733

【書類名】　要約書

【要約】

【課題】　ソケットＡＰＩやＭＰＩ　ＡＰＩを使用した通信の高速化

【解決手段】　５つの新機能を使用する。（１）受信側が、アプリケーション・データ２０２での受信と事前割り振りバッファ２４２での受信のどれが最適かを決定するデータ長を送信側知らせる。（２）アプリケーション・データ２０２の受信アドレスを知らせる効果を計算し、効果が低い場合に知らせを抑える。（３）８つの通信方法を可能にする通信プロトコルを使用する。（４）送受信動作に期待される転送データ長を通信相手にあらかじめ知らせる。（５）通信パターンにより事前割り振りバッファ１４２，２４２を変更する（拡大・縮小・追加・削除等）。

【効果】　通信を高速化し、処理オーバーヘッドとメモリ使用量を減らす。

【選択図】　図１１

認定・付加情報

特許出願の番号	特願２００１－００４３９９
受付番号	５０１００００３２１９２
書類名	特許願
担当官	第七担当上席　　００９６
作成日	平成１３年　１月１５日

＜認定情報・付加情報＞

【提出日】　平成13年 1月12日

次頁無

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所